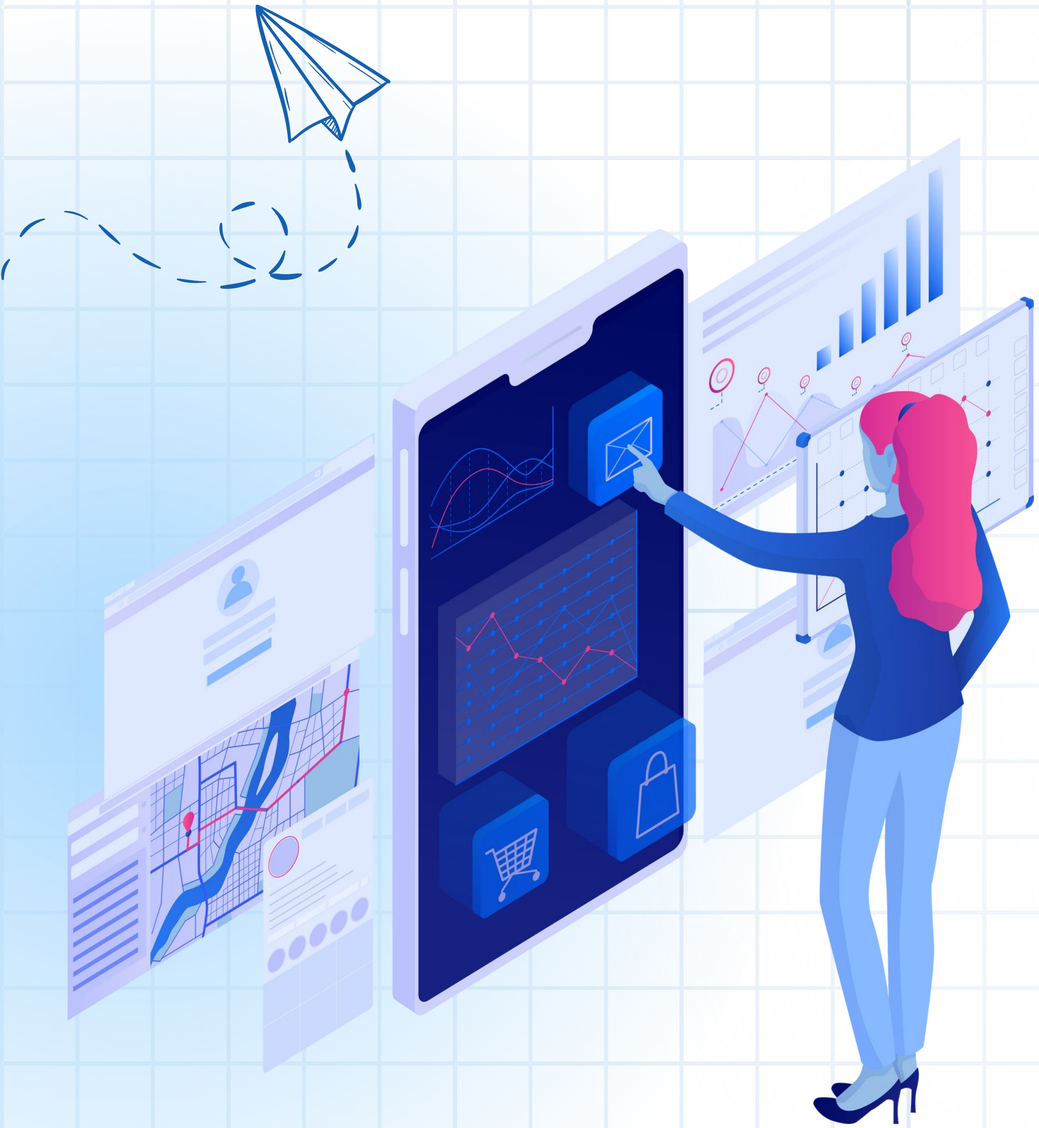
BS./BSc.

in Applied AI and Data Science

Linear algebra and numerical analysis

Powered by Futurense

Module 4: Numerical Linear Algebra
Linear Solvers and Factorization Methods



1. Review of Vector space and linear independence
2. Geometric interpretation of Projection in vector space
3. Concepts of QR decomposition
4. Gram-Schmidt procedure for orthogonalization

Module 4.1: Recap of Orthogonal Matrices

Orthogonal columns

All columns are pair-wise orthogonal.

Unit-norm columns

The norm (geometric length) of each column is exactly 1.

We can translate those two properties into a mathematical expression (remember that $\langle \mathbf{a}, \mathbf{b} \rangle$ is an alternative notation for the dot product):

$$\langle \mathbf{q}_i, \mathbf{q}_j \rangle = \begin{cases} 0, & \text{if } i \neq j \\ 1, & \text{if } i = j \end{cases}$$

Gram-Schmidt procedure

There is a geometric interpretation of the previous example helps in solving algebraic equations

- *The Gram-Schmidt procedure is a way to transform a nonorthogonal matrix into an orthogonal matrix.*
- *Gram-Schmidt has high educational value but, unfortunately, very little application value.*
- *The reason is that—as you've now read several times before—there are numerical instabilities resulting from many divisions and multiplications by tiny numbers.*
- *GS is the right way to conceptualize how and why QR decomposition works even if the low-level implementation is slightly different.*

Gram-Schmidt procedure for orthogonalization

A matrix $\mathbf{V}$ comprising columns $\mathbf{v}_1$ through $\mathbf{v}_n$ is transformed into an orthogonal matrix $\mathbf{Q}$ with columns $\mathbf{q}_k$ according to the following algorithm.

For all column vectors in $\mathbf{V}$ starting from the first (leftmost) and moving systematically to the last (rightmost):

1. Orthogonalize $\mathbf{v}_k$ to all previous columns in matrix $\mathbf{Q}$ using orthogonal vector decomposition. That is, compute the component of $\mathbf{v}_k$ that is perpendicular to $\mathbf{q}_{k-1}$, $\mathbf{q}_{k-2}$, and so on down to $\mathbf{q}_1$. The orthogonalized vector is called $\mathbf{v}_k^*$.[2]
2. Normalize $\mathbf{v}_k^*$ to unit length. This is now $\mathbf{q}_k$, the kth column in matrix $\mathbf{Q}$.

Gram-Schmidt procedure for orthogonalization

Q is obviously different from the original matrix (assuming the original matrix was not orthogonal), so we have lost information about that matrix.

Fortunately, that “lost” information can be easily retrieved and stored in another matrix R that multiplies Q.

3 That leads to the question of how we create R. Creating R is straightforward and comes right from the definition of QR:

$$\begin{aligned} A &= QR \\ Q^T A &= Q^T QR \\ Q^T A &= R \end{aligned}$$

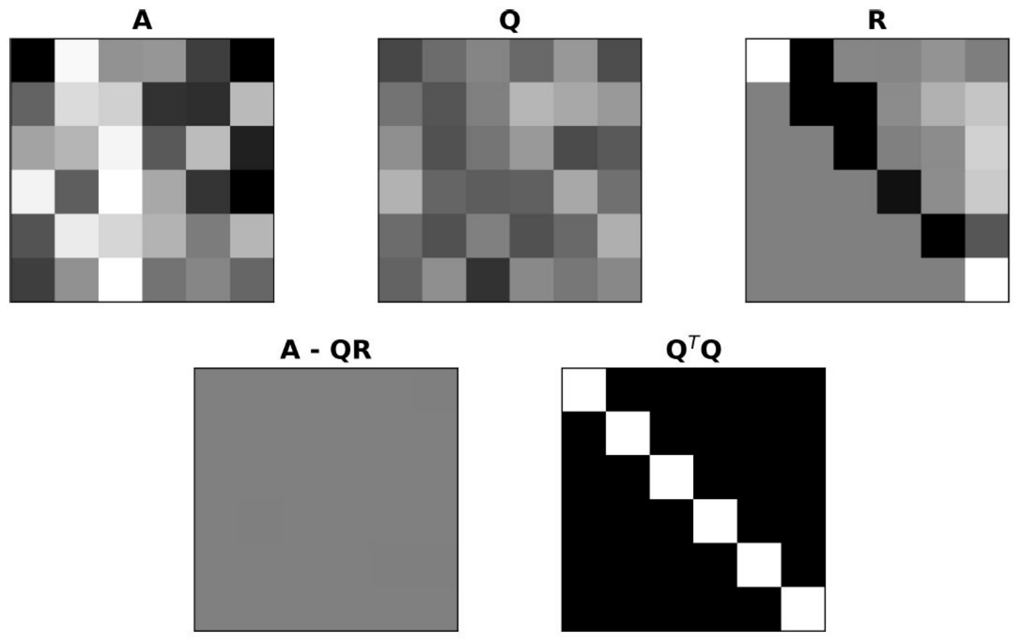
Here you see the beauty of orthogonal matrices: we can solve matrix equations without having to worry about computing the inverse.

Powered by Futurense

Numpy implementation of orthogonalization

```
A = np.random.randn(6,6)
Q,R = np.linalg.qr(A)
```

Several important features of QR decomposition are visible in the following figure



Sizes of Q and R

The sizes of Q and R depend on the size of the to-be-decomposed matrix A and on whether the QR decomposition is “economy” (also called “reduced”) or “full” (also called “complete”).

	A	Q	Q^TQ	QQ^T	R
Square full-rank	MxM r=M	MxM r=M	I_M	I_M	MxM r=M
Square singular	MxM r=k<M	MxM r=M	I_M	I_M	MxM r=k
Tall “full”	M>N r=k	MxM r=M	I_M	I_M	MxN r=k
Tall “economy”	M>N r=k	MxN r=N	I_N	?	MxN r=k
Wide	M<N r=k	MxM r=M	I_M	I_M	MxN r=k

Overview of all possible sizes. Economy versus full QR decomposition applies only to tall matrices. The question is this: for a tall matrix (M > N), do we create a Q matrix with N or M columns? The former option is called economy or reduced and gives a tall Q; the latter is called complete and gives a square Q.

Python example with orthogonalization

Consider the following example in Python:

```
A = np.array([ [1,-1] ]).T
Q,R = np.linalg.qr(A,'complete')
Q*np.sqrt(2) # scaled by sqrt(2) to get integers

>> array([[-1.,  1.],
          [ 1.,  1.]])
```

Because it is possible to craft more than M > N orthogonal vectors from a matrix with N columns, the rank of Q is always the maximum possible rank, which is M for all square Q matrices and N for the economy Q. The rank of R is the same as the rank of A.

Procedure for creating a set of orthonormal vectors through example

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ 3 \end{bmatrix}, \quad \mathbf{v}_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \quad \mathbf{v}_3 = \begin{bmatrix} -2 \\ 1 \end{bmatrix} \quad \Rightarrow \quad \mathbf{V} = \begin{bmatrix} 1 & 1 & -2 \\ 3 & -1 & 1 \end{bmatrix}$$

$$\mathbf{v}_1^* = \frac{1}{\sqrt{10}} \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$

Normalize to unit vector

$$\mathbf{v}_2^* = \mathbf{v}_2 - \mathbf{v}_{2\|\mathbf{v}_1} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} - \frac{1\cdot 1 + 3\cdot -1}{1\cdot 1 + 3\cdot 3} \begin{bmatrix} 1 \\ 3 \end{bmatrix} = \begin{bmatrix} 1 + 1/5 \\ -1 + 3/5 \end{bmatrix} = \frac{1}{5} \begin{bmatrix} 6 \\ -2 \end{bmatrix}$$

Orthogonalize to $\mathbf{v}_1$

$$\mathbf{v}_2^* = \frac{\sqrt{5}}{10\sqrt{2}} \begin{bmatrix} 6 \\ -2 \end{bmatrix}$$

Normalize to unit vector

Procedure for creating a set of orthonormal vectors through example

$$\mathbf{v}_3^* = \mathbf{v}_3 - \mathbf{v}_{3\|\mathbf{v}_1^*} - \mathbf{v}_{3\|\mathbf{v}_2^*} = \begin{bmatrix} -2 \\ 1 \end{bmatrix} - \frac{10^{-1/2} \cdot -2 + 3 \cdot 10^{-1/2}}{(10^{-1/2})^2 + (3 \cdot 10^{-1/2})^2} \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$

$$- \frac{6\sqrt{5}/10\sqrt{2} \cdot -2 + -2\sqrt{5}/10\sqrt{2}}{(6\sqrt{5}/10\sqrt{2})^2 + (-2\sqrt{5}/10\sqrt{2})^2} \begin{bmatrix} 6 \\ -2 \end{bmatrix}$$

$$= \begin{bmatrix} -2 \\ 1 \end{bmatrix} - \frac{10^{-1/2}}{\sqrt{10}} \begin{bmatrix} 1 \\ 3 \end{bmatrix} - \sqrt{5}\frac{\frac{-7\sqrt{5}}{5\sqrt{2}}}{10\sqrt{2}} \begin{bmatrix} 6 \\ -2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Orthogonalize to $\mathbf{v}_1^*$ and to $\mathbf{v}_2^*$

Putting these vectors into a matrix yields

$$\mathbf{Q} = \begin{bmatrix} \frac{1}{\sqrt{10}} & \frac{3\sqrt{5}}{5\sqrt{2}} & 0 \\ \frac{3}{\sqrt{10}} & \frac{3\sqrt{5}}{10\sqrt{2}} & 0 \end{bmatrix}$$

Question

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad \Rightarrow \quad \mathbf{Q} = \begin{bmatrix} \sqrt{2}^{-1} & \sqrt{6}^{-1} \\ \sqrt{2}^{-1} & -\sqrt{6}^{-1} \\ 0 & -2\sqrt{6}^{-1} \end{bmatrix}$$

Given the above vectors, how can you compute the R matrix?

Solution

$$\mathbf{R} = \mathbf{Q}^{\mathrm{T}}\mathbf{A} = \begin{bmatrix} \sqrt{2}^{-1} & \sqrt{2}^{-1} & 0 \\ \sqrt{6}^{-1} & -\sqrt{6}^{-1} & -2\sqrt{6}^{-1} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \sqrt{2} & \sqrt{2}^{-1} \\ 0 & -\sqrt{6}/2 \end{bmatrix}$$

R is an upper triangular matrix. That's not just some happy coincidence. R will always be an upper triangular matrix.

Module 4.2: Why R is upper triangular matrix

I hope you gave this question some serious thought. It's a tricky point about QR decomposition, so if you couldn't figure it out on your own

I will start by reminding you of three facts:

- $\mathbf{R}$ comes from the formula $\mathbf{Q}^T\mathbf{A} = \mathbf{R}$.
- The lower triangle of a product matrix comprises dot products between *later* rows of the left matrix and *earlier* columns of the right matrix.
- The rows of $\mathbf{Q}^T$ are the columns of $\mathbf{Q}$.

QR and inverses

- QR decomposition provides a more numerically stable way to compute the matrix inverse.
- Let's start by writing out the QR decomposition formula and inverting both sides of the equation (note the application of the LIVE EVIL rule):

$$
\begin{aligned}
A &= QR \\
A^{-1} &= (QR)^{-1} \\
A^{-1} &= R^{-1}Q^{-1} \\
A^{-1} &= R^{-1}Q^{T}
\end{aligned}
$$

QR and inverses

- Thus, we can obtain the inverse of A as the inverse of R times the transpose of Q. Q is numerically stable due to the Householder reflection algorithm, and R is numerically stable because it simply results from matrix multiplication.

- We still need to invert R explicitly, but inverting triangular matrices is highly numerically stable when done through a back substitution procedure. You'll learn more about that in the next chapter, but the key point is this: an important application of QR decomposition is providing a more numerically stable way to invert matrices

Question 3

Implement the Gram-Schmidt Procedure procedure to orthonormalize the following set of vectors in $\mathbb{R}^3$ and calculate a set of orthonormal basis vectors:

- $v_1 = (1, 1, 0)$
- $v_2 = (1, 2, 0)$
- $v_3 = (0, 1, 2)$

Solution

Orthogonalization:

- $u_1 = v_1 = (1, 1, 0)$ $u_2 = v_2 - ((v_2 \cdot u_1) / (u_1 \cdot u_1)) * u_1$ $v_2 \cdot u_1 = (1)(1) + (2)(1) + (0)(0) = 3$
- $u_1 \cdot u_1 = (1)(1) + (1)(1) + (0)(0) = 2$
- $u_2 = (1, 2, 0) - (3/2)(1, 1, 0) = (-1/2, 1/2, 0)$
- $u_3 = v_3 - ((v_3 \cdot u_1) / (u_1 \cdot u_1)) * u_1 - ((v_3 \cdot u_2) / (u_2 \cdot u_2)) * u_2$ $v_3 \cdot u_1 = (0)(1) + (1)(1) + (2)(0) = 1$
- $v_3 \cdot u_2 = (0)(-1/2) + (1)(1/2) + (2)(0) = 1/2$
- $u_2 \cdot u_2 = (-1/2)(-1/2) + (1/2)(1/2) + (0)(0) = 1/2$
- $u_3 = (0, 1, 2) - (1/2)(1, 1, 0) - (1)(-1/2, 1/2, 0) = (0, 0, 2)$

Normalization:

- $||u_1|| = \sqrt{(1^2 + 1^2 + 0^2)} = \sqrt{2}$
- $e_1 = (1/\sqrt{2}, 1/\sqrt{2}, 0)$
- $||u_2|| = \sqrt{((-1/2)^2 + (1/2)^2 + 0^2)} = \sqrt{(1/2)} = 1/\sqrt{2}$
- $e_2 = (-1/\sqrt{2}, 1/\sqrt{2}, 0)$
- $||u_3|| = \sqrt{(0^2 + 0^2 + 2^2)} = 2$
- $e_3 = (0, 0, 1)$

Therefore, the orthonormal basis is:

- $e_1 = (1/\sqrt{2}, 1/\sqrt{2}, 0)$
- $e_2 = (-1/\sqrt{2}, 1/\sqrt{2}, 0)$
- $e_3 = (0, 0, 1)$

Recap

QR decomposition is great. It's definitely on my list of the top five most awesome matrix decompositions in linear algebra. QR decomposition is great. It's definitely on my list of the top five most awesome matrix decompositions in linear algebra.

- •• An orthogonal matrix has columns that are pair-wise orthogonal and with norm = 1. Orthogonal matrices are key to several matrix decompositions, including QR, eigen decomposition, and singular value decomposition. Orthogonal matrices are also important in geometry and computer graphics (e.g., pure rotation matrices).

- •• You can transform a nonorthogonal matrix into an orthogonal matrix via the Gram-Schmidt procedure, which involves applying an orthogonal vector decomposition to isolate the component of each column that is orthogonal to all previous columns ("previous" meaning left to right).

** QR decomposition is the result of Gram-Schmidt (technically, it is implemented by a more stable algorithm, but GS is still the right way to understand it).

Coming up next...............

- Sparse matrices
- Numerical linear algebra software
- Linear Solvers and Factorization Methods

Thank you